Doc Code: AP.PRE.REQ

OPAP
MAY 1 8 2010
IAP16
PATENT & TRADEMARK OFFICE

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PRE-APPEAL BRIEF REQUEST FOR REVIEW		**Docket Number (Optional)** JRL-4147-144
	Application Number 10/571,606	Filed March 10, 2006
	First Named Inventor MEIRICK	
	Art Unit 2617	Examiner Patel, Mahendra R.

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ Applicant/Inventor

☐ Assignee of record of the entire interest. See 37 C.F.R. § 3.71. Statement under 37 C.F.R. § 3.73(b) is enclosed. (Form PTO/SB/96)

☒ Attorney or agent of record 33,149 (Reg. No.)

☐ Attorney or agent acting under 37CFR 1.34.
Registration number if acting under 37 C.F.R. § 1,34 ______

Signature

John R. Lastova
Typed or printed name

703-816-4025
Requester's telephone number

May 18, 2010
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.*

☒ *Total of 1 form/s are submitted.

1635043

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

MEIRICK ET AL. — Atty. Ref.: 4147-144; Confirmation No. 8029

Appl. No. 10/571,606 — TC/A.U. 2617

Filed: March 10, 2006 — Examiner: Patel, Mahendra R.

For: METHOD FOR DISCARDING ALL SEGMENTS CORRESPONDING TO THE SAME PACKET IN A BUFFER

* * * * * * * * * * *

May 18, 2010

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Besset discloses a method of managing a data buffer comprising a queue of consecutive segments of data packets in a base station system of a communications system. See the Abstract; 1:6-8, and 4:33-37. A base station system identifies a complete data packet (AAL2 SDU) based on a user-to-user interface (UUI) field included in CPS packet segments. See 2:59-63; 5:10-15, 26-29, 40-43; 6:1-20; 9:27-30; 10:31-42. The base station system discards the identified complete data packet. See 4:41-43; 5:19-23, 57-61; and 6:8-20.

Clear Error #1: Besset fails to disclose that the base station system compares a size of a data packet segment with a size of a next consecutive data packet segment in the buffer. Besset reads the Length Indicator (LI) field of the CPS Packet header to determine the length of an arriving CPS Packet (6:31-33). The determined length is used to update the buffer's current filling level, i.e., CPS_CO+LI+1+3, where CPS_CO indicates the buffer's current filling level, LI is the length indicator, 3 is the size of the header, and LI+1 is the size of the CPS packet payload (8:37-38, 51-60). The updated buffer's filling level is then compared to a threshold (CPS_Low_Threshold) to decide whether the buffer is in a state of congestion (8:60-67).

Contrary to the Examiner's contentions, Besset does not disclose comparing the sizes of consecutive data packet segments. Using the example in the Advisory Action, when a first data

packet segment is received, the UUI field of the packet segment is first investigated to determine whether it is the first packet segment of an AAL2 SDU (8:23-28). Second, the length of the data packet segment is determined by reading the LI field (9:32-34; 8:51-67). Third, the buffer's current filling level CPS_CO-value updated with the length of the data packet is then compared to a buffer level lower threshold, i.e., 0+LI+1+3 is compared to the CPS_Low_Threshold (9:32-34; 8:51-67). Because the buffer was empty (CPS_CO=0) and there is no congestion in this example, the data packet segment is entered in the buffer (9:37-40), and the buffer's current filling level CPS_CO is updated to be CPS_CO+LI+1+3 (9:40-42). Assuming that the next data packet arriving at the buffer is the next consecutive (the second) data packet segment, these three steps are repeated for that data packet segment. The updated CPS_CO value is thus LI (first data packet segment)+1+3+LI (second data packet segment)+1+3. This updated CPS_CO-value is compared to CPS_Low_Threshold.

A key distinction between what is claimed and what Besset describes is that Besset compares the **sum** of the lengths of the first data packet segment and the second data packet segment to a buffer level **threshold**. Besset does not compare the **size** of the first data packet segment with the **size** of the next data packet segment as claimed. Comparing the sum of data packet segment sizes with a buffer level threshold is not the same and does not give the same comparison result as comparing the sizes of two consecutive data packet segments.

Another missing feature overlooked by the Examiner is that claim 1 recites that the packet segment size comparison is for data packet segments "**in the buffer.**" The preamble of claim 1, a "method of managing a data buffer comprising a queue of consecutive segments of data packets" gives the context for "comparing a size of a data packet segment with a size of a next consecutive data packet segment **in said buffer**." In contrast, Besset uses the determined length of a data packet segment to update the buffer's current filling level CPS_CO-value and compares this updated CPS_CO-value with the threshold **prior** to deciding whether to enter the data packet segment **in the buffer** (9:32-40).

Clear Error #2: Besset fails to teach "said base station system identifying a complete data packet in said buffer based on said comparison [of the sizes of consecutive segments in the buffer]." The UUI field of Besset's data packet segments is used to identify whether the end of an SSAR SDU has been reached (UUI field =26) or more data packet segments follows (UUI field=27) (6:1-20; 10:35-42). In the Advisory, the Examiner contends that when Besset's buffer

is empty, the first data packet segment is stored, and when the second data packet arrives, the algorithm compares the updated CPS_CO-value with the CPS_Low_Threshold to determine the congestion level. The Examiner concludes: "And from header, it will determine if last packet is arrived before reassembling."

So the Examiner admits that Besset uses the header information, i.e., the UUI field, to determine whether the last data packet has arrived, thereby allowing identification of a complete data packet. But Besset's comparison of the buffer's current filling level CPS_CO-value with the CPS-Low_Threshold, (i.e., the comparison relied on by the Examiner for the first comparing step of claim 1), has nothing to do with any complete data packet identification and is instead solely used to monitor buffer congestion. The contradictory positions taken by the Examiner—Besset uses header information to identify a complete data packet (true) v. Besset identifies of a complete data packet based on packet size comparison (not true)—constitute clear error. The only identification of a complete data packet disclosed by Besset is based on the UUI fields of the data packet segments (6:1-6, 13-15; 10:35-41).

In addition, Besset discloses reading the UUI field of an arrived data packet segment **prior** to it being entered in the buffer (9:23-30). As a result, Besset fails to teach identification of a complete data packet **in the buffer** based on the segment size comparison.

Clear Error #3: Besset fails to teach "said base station system discarding said identified complete data packet from said buffer." Besset discloses that if a first data packet segment of a complete data packet is entered in the buffer, then all remaining data packet segments will be entered in the buffer, even if that causes congestion (6:9-13; 9:64-67)). But if the buffer is already congested when the first data packet segment of a complete data packet arrives at the buffer, then that data packet segment is not entered in the buffer; nor are any of the other following data packet segments for that complete data packet (10:5-20).

The Examiner's example at (2) in the Advisory presents a first data packet segment arriving at and entered into an empty buffer. The Examiner assumes that the overflow level of the buffer is set so that arrival of the second data packet segment causes overflow, with congestion being detected. The Examiner concludes that Besset discards these two data packet segments. This is conclusion is incorrect. Besset explains that once a first segment of a complete data packet is in the buffer, **all** consecutive data packet segments belonging to that complete data packet are also entered in the buffer without any congestion determination (9:64-

67). So the situation envisioned by the Examiner, i.e., entering a first data packet segment, and then deciding, when receiving the second data packet segment, that congestion has occurred and discarding both data packet segments will **never** occur according to Besset.

In contrast, either all data packet segments of a complete data packet are stored in the buffer or none of the data packet segments of a complete data packet is entered in the buffer, as disclosed by Besset. Besset's base station system preventing entry of the identified complete data packet in the buffer does not disclose discarding any identified complete data packet from the buffer.

Clear Error #4: The combination of Besset and Yuan does not disclose discarding an identified complete data packet from a buffer. Yuan merely prevents a complete data packet segment from entering the buffer. As a result, the complete data packet segment cannot be discarded from the buffer because it was never stored in the buffer. Both Besset and Yuan identify segments by retrieving header information, i.e., the CPS UUI field in the header of CPS packets to identify a complete AAL2 SDU frame versus first cell identifier of ATM cell header (Besset) and decrementing counter field in the header CRC field (Yuan). Consequently, neither reference discloses or suggests comparing sizes of data packet segments in order to identify a complete data packet. To the contrary, the combined teachings of Besset and Yuan direct the person skilled in the art to use header information in order to identify the data packet segments belonging to a complete data packet—a different approach from what is claimed.

Like Besset, Yuan also decides whether to enter segments in a buffer or discard segments prior to entry in the buffer. Yuan discloses that a controller decides whether the buffer contains sufficient space to store received cells, allows storage of the received cells when the buffer contains sufficient space, and discards the received cells when the buffer contains insufficient space (2:11-18, 51-53; 5:43-49). Yuan's discard mechanism is the same as Besset's. If the first cell is accepted and entered in the buffer, then all remaining cells of the complete data packet are also entered in the buffer. But if the first cell is not entered in the buffer, then all cells of that complete data packet are discarded and never enter the buffer (6:5-8). Thus, Besset and Yuan discard a complete data packet prior to entering the segments/cells in the buffer. So the combination fails to teach that an identified complete data packet is discarded from a buffer that already contains the segments of the data packet.

The clear errors noted above for claim 1 also apply to claims 5, 10, and 20. The final rejection should be withdrawn and the case allowed.

Respectfully submitted,
NIXON & VANDERHYE P.C.

By: ______________________
John R. Lastova
Reg. No. 33,149

901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000